Complexity reduction for high-dimensional kinetic equations

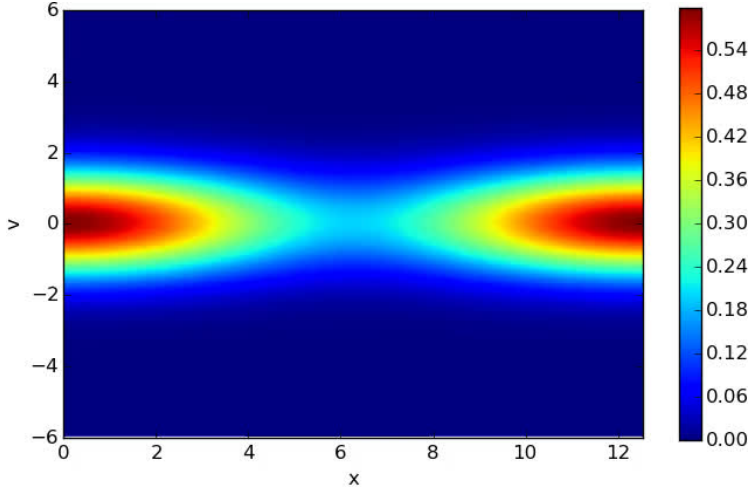
Lukas Einkemmer
University of Innsbruck

Dobbiaco summer school, 2022

Link to slides: `http://www.einkemmer.net/training.html`

Nonlinear Landau damping

Small scales in phase space are a common feature of kinetic simulation.



Numerical challenges

Small scale structures force a sufficiently fine space discretization.

The phase space is **up to six-dimensional**.

- $n = 50$ 250 GB memory (workstation)
- $n = 100$ 16 TB memory (local cluster)
- $n = 200$ 1024 TB memory (largest supercomputer)

We need a numerical method

- for which stability is not dictated by $v\tau < h$
- **that does not introduce additional memory requirements**
- **that is scalable to large HPC systems**
- that does not introduce too much numerical diffusion

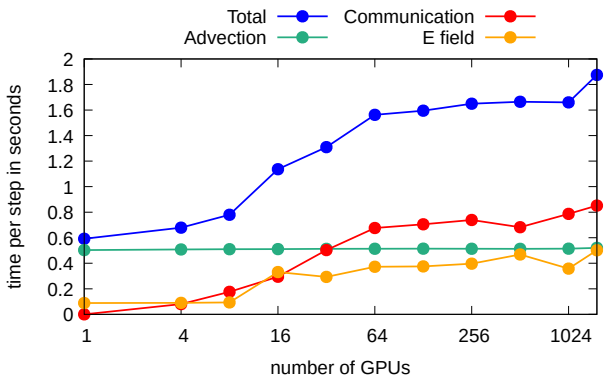
High-performance computing

To obtain results for **five or six-dimensional** problems requires the **largest supercomputers currently available** (perhaps more than that).

Simulation using ≈ 1500 GPUs and $72^3 144^3$ grid points.

JUWELS Booster:

- 2×24 AMD EPYC 7402 cores and $4\times$ NVIDIA A100 GPUs per node.
- Total of 150 TB of GPU memory.
- $4\times$ Mellanox HDR200 InfiniBand ConnectX 6 (200 Gbit/s each).



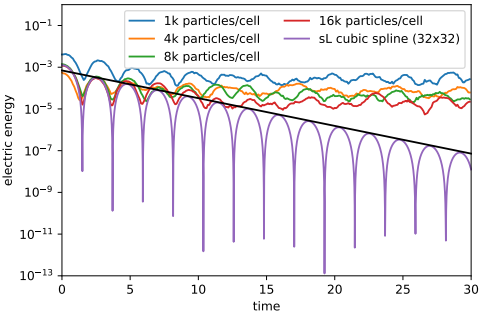
Dimension reduction for the Vlasov equation

Fundamental problem of Eulerian Vlasov solvers is that effort scales as $\mathcal{O}(n^{d_x+d_v})$.

- **Curse of dimensionality**

Particle methods have been employed extensively.

- Only x is discretized.
- Particles push and field solves are alternated.



Sparse grids

- Have problems resolving Gaussians.
- regularity is an issue as $\|\partial_v^m f(t,\cdot,\cdot))\| \propto t^m$.

E. Camporeale et al., 198, Comput. Phys. Commun., 2016.
K. Kormann, E. Sonnendrücker, Sparse Grids and Applications. 2016.

Dynamical low-rank approximation

Singular value decomposition

Singular value decomposition for a matrix $G_{ij} = g(x_i, v_j)$ is given by

$$G = VSW^T \in \mathbb{R}^{n \times m}$$

with $V \in \mathbb{R}^{n \times r}$, $S \in \mathbb{R}^{r \times r}$, $W \in \mathbb{R}^{m \times r}$, and r the rank of G.

Low-rank approximation

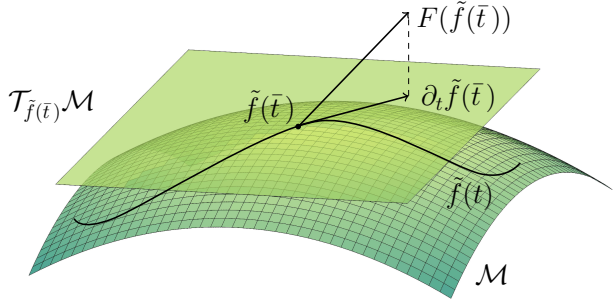
$$g(x, v) \approx \sum_{ij} X_i(x) S_{ij} V_j(v).$$

Orthogonality constraints: $\langle X_i, X_j \rangle_x = \delta_{ij}$, $\langle V_i, V_j \rangle_v = \delta_{ij}$.

Why do we think that low-rank works any better?

Dynamical low-rank approximation

Dynamical low-rank approximation



Advantages: method does not leave manifold, analyze DLR without discretization, better stability properties.

O. Koch, C. Lubich. SIAM J. Matrix Anal. Appl., 29(2), 2007.
J. Kusch, L.E., G. Ceruti. arXiv:2107.07282.

Dynamical low-rank approximation

Dynamical low-rank approximation

$$f(t,x,v) = \sum_{ij} X_i(t,x) S_{ij}(t) V_j(t,v).$$

Low-rank functions (with fixed r) form a **manifold** with functions in the tangent space represented as

$$\dot{f} = \sum_{ij} \left(\dot{X}_i S_{ij} V_j + X_i \dot{S}_{ij} V_j + X_i S_{ij} \dot{V}_j \right).$$

This representation is **not unique**. For example,

$$\dot{X}_i = X_i,\ \dot{S}_{ij} = 0 \qquad \text{and} \qquad \dot{X}_i = 0,\ \dot{S}_{ij} = S_{ij}$$

gives the same vector in the tangent space.

Gauge conditions

We impose the **Gauge conditions** $\langle X_i, \dot{X}_j \rangle_x = 0$ and $\langle V_i, \dot{V}_j \rangle_v = 0$.

Equation for S

$$\begin{aligned}\langle X_k V_l, \dot{f} \rangle &= \sum_{ij} \langle X_k V_l, \dot{X}_i S_{ij} V_j + X_i \dot{S}_{ij} V_j + X_i S_{ij} \dot{V}_j \rangle_{xv} \\ &= \sum_{ij} \langle X_k, \dot{X}_i \rangle_x S_{ij} \langle V_l, V_j \rangle_v + \sum_{ij} \langle X_k, X_i \rangle_x \dot{S}_{ij} \langle V_l, V_j \rangle_v + \sum_{ij} \langle X_k, X_i \rangle_x S_{ij} \langle V_l, \dot{V}_j \rangle_v \\ &= \dot{S}_{kl}\end{aligned}$$

Equation for X

$$\begin{aligned}\langle V_l, \dot{f} \rangle &= \sum_{ij} \langle V_l, \dot{X}_i S_{ij} V_j + X_i \dot{S}_{ij} V_j + X_i S_{ij} \dot{V}_j \rangle_{xv} \\ &= \sum_{ij} \dot{X}_i S_{ij} \langle V_l, V_j \rangle_v + \sum_{ij} X_i \dot{S}_{ij} \langle V_l, V_j \rangle_v + \sum_{ij} X_i S_{ij} \langle V_l, \dot{V}_j \rangle_v \\ &= \sum_i \dot{X}_i S_{il} + \sum_i X_i \dot{S}_{il}\end{aligned}$$

Dynamical low-rank approximation

Equations of motion

$$\partial_t S_{ij} = \langle X_i V_j, \mathrm{RHS} \rangle, \qquad \text{ODE}$$

$$\sum_i S_{ij} (\partial_t X_i) = \langle V_j, \mathrm{RHS} \rangle - \sum_i X_i (\partial_t S_{ij}), \qquad \text{x dependent PDE}$$

$$\sum_j S_{ij} (\partial_t V_j) = \langle X_i, \mathrm{RHS} \rangle - \sum_j (\partial_t S_{ij}) V_j. \qquad \text{v dependent PDE}$$

In principle we can substitute

$$\mathrm{RHS} = -v \cdot \nabla_x f + E(f) \cdot \nabla_v f.$$

But

- The equations couple S, X, and V;
- To obtain equations in X_i and V_j we have to invert S and S^{T}.

Robustness

Back to the SVD

$$A \approx VSW^T.$$

Approximation by truncation

$$S = \begin{bmatrix} \mu_1 & 0 & 0 & 0 & 0 \\ 0 & \mu_2 & 0 & 0 & 0 \\ 0 & 0 & \mu_3 & 0 & 0 \\ 0 & 0 & 0 & \mu_4 & 0 \\ 0 & 0 & 0 & 0 & \mu_5 \end{bmatrix} \approx \begin{bmatrix} \mu_1 & 0 & 0 \\ 0 & \mu_2 & 0 \\ 0 & 0 & \mu_3 \end{bmatrix}.$$

Error vs condition number

- If μ_4 is large than the error is large.
- If μ_3 is small than inverting S is ill-conditioned.

Projector splitting integrator

Vlasov–Poisson equation **constrained to the low-rank manifold**

$$\partial_t f = P(f)\mathrm{RHS} = P(f)\left(-v \cdot \nabla_x f + E(f) \cdot \nabla_v f\right),$$

where $P(f)$ is the orthogonal projector onto the tangent space.

We have

$$\begin{aligned} P(f)\mathrm{RHS} &= \sum_{ij} \left(\partial_t X_i S_{ij} V_j + X_i \partial_t S_{ij} V_j + X_i S_{ij} \partial_t V_j\right) \\ &= \sum_{ij} \left(\partial_t (X_i S_{ij}) V_j - X_i \partial_t S_{ij} V_j + X_i \partial_t (S_{ij} V_j)\right) \\ &= \sum_j \langle V_j, \mathrm{RHS} \rangle_x V_j - \sum_{ij} X_i \langle X_i V_j, \mathrm{RHS} \rangle_{xv} V_j + \sum_i X_i \langle X_i, \mathrm{RHS} \rangle_v \end{aligned}$$

Projector splitting integrator

We can write

$$P(f)g = P_{\overline{V}}g - P_{\overline{V}}P_{\overline{X}}g + P_{\overline{X}}g,$$

where $P_{\overline{X}}$ and $P_{\overline{V}}$ are the orthogonal projectors on $\overline{X} = \text{span}\{X_i \colon i = 1 \dots r\}$ and $\overline{V} = \text{span}\{V_j \colon j = 1 \dots r\}$.

This suggests a **splitting**.

K step

Our goal is to solve

$$\partial_t f = P_{\overline{V}}\left(-v \cdot \nabla_x f + E(f) \cdot \nabla_v f\right).$$

We rewrite the solution using K_j as follows

$$f(t,x,v) = \sum_j K_j(t,x) V_j(t,v), \qquad \text{with} \quad K_j(t,x) = \sum_i X_i(t,x) S_{ij}(t).$$

This yields

$$\sum_j \partial_t K_j(t,x) V_j(t,v) + \sum_j K_j(t,x) \partial_t V_j(t,v)$$
$$= \sum_j \left\langle V_j(t,\cdot), v \mapsto -v \cdot \nabla_x f(t,x,v) + E(f)(t,x) \cdot \nabla_v f(t,x,v) \right\rangle_v V_j(t,v).$$

K step

The solution is given by $V_j(t,v) = V_j(0,v)$ and

$$\begin{aligned}\partial_t K_j(t,x) &= \langle V_j, -v \mapsto v \cdot \nabla_x f(t,x,v) + E(f)(t,x) \cdot \nabla_v f(t,x,v) \rangle_v \\ &= -\sum_l \langle V_j v V_l \rangle_v \cdot \nabla_x K_l(t,x) + \sum_l E \cdot \langle V_j \nabla_v V_l \rangle_v K_l(t,x)\end{aligned}$$

For the **first subflow of the projector splitting algorithm** we thus obtain

$$\partial_t K_j(t,x) = -\sum_l c^1_{jl} \cdot \nabla_x K_l(t,x) + \sum_l c^2_{jl} \cdot E(K)(t,x) K_l(t,x),$$

The coefficients are determined as follows ($V = V^0$)

$$c^1_{jl} = \int_{\Omega_v} v V^0_j V^0_l \,\mathrm{d}v, \quad c^2_{jl} = \int_{\Omega_v} V^0_j (\nabla_v V^0_l) \,\mathrm{d}v.$$

Do not neglect the cost of computing the coefficients.

K step

The equation is formulated with K and V (neither X nor S are explicitly involved).

To proceed with the next step in the algorithm we have to obtain X and S.

- **Why is this approach then advantageous?**

The X and S are recovered from K by a **QR decomposition** as

$$K_j = \sum_i X_i S_{ij}$$

Well defined even for singular $K = [K_1, \ldots, K_r]$ and gives automatically the (almost correct) orthogonality relation for the X_i.

- Result is a robust approximation even if the rank r is chosen *too large*.

Note that S is not necessarily diagonal.

S step

Our goal is to solve

$$\partial_t f = -P_V P_X \left(-v \cdot \nabla_x f + E(f) \cdot \nabla_v f\right).$$

The solution is $X_i(t,x) = X_i(0,x)$, $V_j(t,v) = V_j(0,v)$, and

$$\begin{aligned}\partial_t S_{ij} &= \langle X_i^1 V_j^0, (x,v) \mapsto (v \cdot \nabla_x - E(S)(t,x) \cdot \nabla_v) \sum_{kl} X_k^1(x) S_{kl}(t) V_l^0(v) \rangle_{xv} \\ &= \sum_{kl} \left(c_{jl}^1 \cdot d_{ik}^2 - c_{jl}^2 \cdot d_{ik}^1[E(S(t))] \right) S_{kl}(t)\end{aligned}$$

with

$$d_{ik}^1[E] = \int_{\Omega_x} X_i^1 E X_k^1 \,\mathrm{d}x, \qquad d_{ik}^2 = \int_{\Omega_x} X_i^1 (\nabla_x X_k^1) \,\mathrm{d}x.$$

The S step integrates backward in time.

Electric field (S step)

The electric field $E(S(t))$ is computed from

$$-\Delta\phi = 1 - \sum_{ij} X_i^1(x) S_{ij}(t) \int V_j^0 \, \mathrm{d}v, \qquad E = -\nabla\phi.$$

In practice we usually approximate E by E^n (first order) or $E^{n+1/2}$ (second order).

- $E^{n+1/2}$ has to be approximated (to first order) in an actual implementation.

L step

Our goal is to solve

$$\partial_t f = P_{\overline{X}}\left(-v \cdot \nabla_x f + E(f) \cdot \nabla_v f\right).$$

We define

$$f(t,x,v) = \sum_i X_i(t,x) L_i(t,v), \qquad \text{with} \quad L_i(t,v) = \sum_j S_{ij}(t) V_j(t,v).$$

The solution is $X_i(t,x) = X_i(0,x)$ and

$$\begin{aligned}\partial_t L_i(t,v) &= \left\langle X_j^1, x \mapsto (-v \cdot \nabla_x + E(L)(t,x) \cdot \nabla_v) \sum_k X_k^1 L_k(t,v) \right\rangle_x \\ &= \sum_k d_{ik}^1 [E(L(t,\cdot))] \cdot \nabla_v L_k(t,v) - \sum_k (d_{ik}^2 \cdot v) L_k(t,v).\end{aligned}$$

Then S and V are recovered from L by a **QR decomposition**.

Dynamical low-rank algorithm

First order Lie splitting

1. Solve $\partial_t K_j = -\sum_l c^1_{jl} \cdot \nabla_x K_l + \sum_l c^2_{jl} \cdot E(K) K_l$ with initial value $\sum_i X^0_i S^0_{ij}$ up to time Δt to obtain K^1_j.

2. Perform a QR decomposition of K^1_j to obtain X^1_i and S^1_{ij}.

3. Solve $\partial_t S_{ij} = \sum_{kl} \left(c^1_{jl} \cdot d^2_{ik} - c^2_{jl} \cdot d^1_{ik} \right) S_{kl}$ with initial value S^1_{ij} up to time Δt to obtain S^2_{ij}.

4. Solve $\partial_t L_i = \sum_k d^1_{ik} \cdot \nabla_v L_k - \sum_k (d^2_{ik} \cdot v) L_k$ equation with initial value $\sum_j S^2_{ij} V^0_j$ up to time Δt to obtain L^1_i.

5. Perform a QR decomposition of L^1_i to obtain V^1_j and S^3_{ij}.

Spectral and **semi-Lagrangian** methods can still be employed.

L.E., C. Lubich, SIAM J. Sci. Comput. 40(5), 2018.

Computational complexity

Computational complexity: $\mathcal{O}(r^2 n^d)$ instead of $\mathcal{O}(n^{d_x+d_v})$.

- Limited by computation of the coefficients and solution of evolution equations.

Memory usage: $\mathcal{O}(rn^d)$ instead of $\mathcal{O}(n^{d_x+d_v})$.

- Limited by storage of X_i and V_j.

Coefficients:	$c_{jl}^1 = \int_{\Omega_v} v V_j^0 V_l^0 \,\mathrm{d}v$	Storage: $\mathcal{O}(r^2)$	Effort: $\mathcal{O}(r^2 n^{d_v})$
Integration:	$\partial_t K_j = \ldots$	Storage: $\mathcal{O}(rn^{d_x})$	Effort: $\mathcal{O}(r^2 n^{d_x})$

$d = \max(d_x, d_v)$.

Implementation

Discretized system

$$f = XSV^{\mathsf{T}}$$

with

$$f_{kl} = f(t, x_k, v_l), \qquad X_{ki} = X_i(t, x_k), \qquad V_{lj} = V_j(t, v_l).$$

In **matrix form**

$$X(t) = \begin{bmatrix} X_1(t, x_1) & \cdots & X_r(t, x_1) \\ \vdots & \ddots & \vdots \\ X_1(t, x_n) & \cdots & X_r(t, x_n) \end{bmatrix}, \qquad V(t) = \begin{bmatrix} V_1(t, v_1) & \cdots & V_r(t, v_1) \\ \vdots & \ddots & \vdots \\ V_1(t, v_m) & \cdots & V_r(t, v_m) \end{bmatrix}.$$

K step (one-dimensional case)

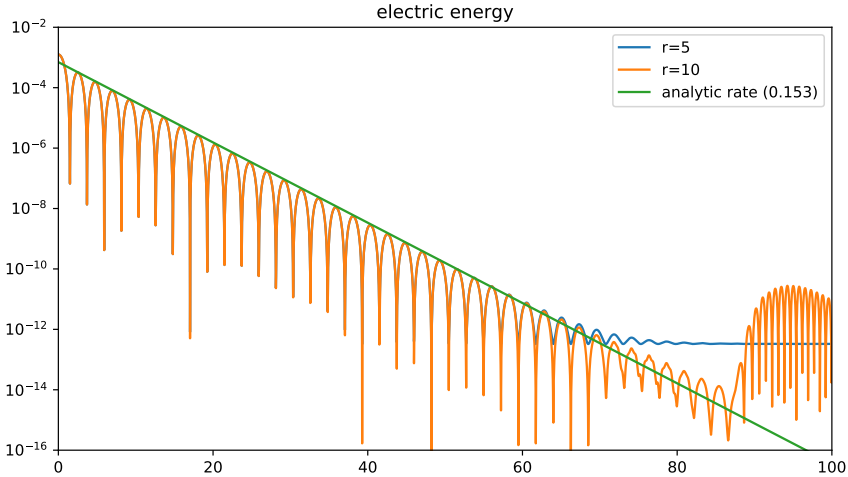
$$\partial_t K = -A_{\partial_x} K (c^1)^T + \mathrm{diag}(E^n) K (c^2)^T,$$

where A_{∂_x} is the discretization of the spatial derivative.

Why does dynamical low-rank work?

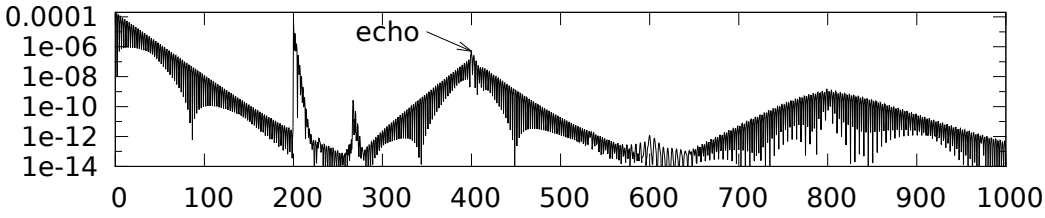
Linear Landau damping

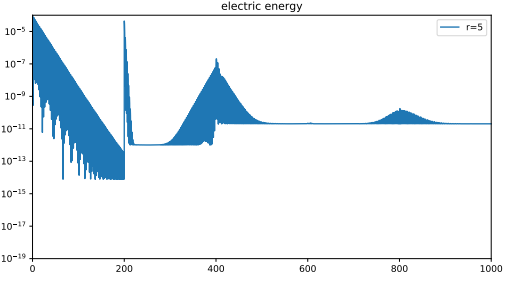
Low-rank approximation with 256 grid points in each direction.

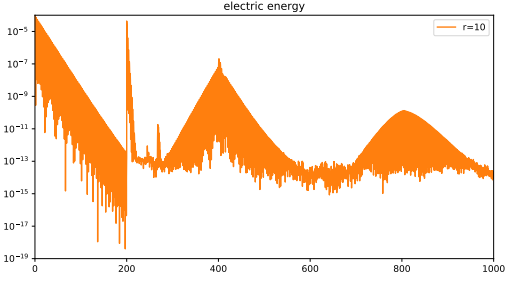


Plasma echo

Plasma echo with 512×4096 grid points.







Reason 1: DLR does not need smoothness

The low-rank algorithm is able to **resolve filamentation**. Consider

$$\partial_t f(t,x,v) + v \cdot \nabla_x f(t,x,v) = 0, \qquad f(0,x,v) = \mathrm{e}^{ikx}\mathrm{e}^{-v^2}.$$

Then

$$f(t,x,v) = \mathrm{e}^{ik(x-vt)}\mathrm{e}^{-v^2} = \mathrm{e}^{ikx}\mathrm{e}^{-ikvt}\mathrm{e}^{-v^2}.$$

This is still rank 1.

Thus, **smoothness in v is not necessary** for low-rank approximations.

Reason 2: DLR resolves dynamics close to the linear regime

We consider a small perturbation around the equilibrium $f^{(0)}(v)$

$$f(t,x,v) = f^{(0)}(v) + f^{(1)}(t,x,v), \qquad E(t,x) = 0 + E^{(1)}(t,x).$$

This results in the **linearized Vlasov equation**

$$\partial_t f^{(1)}(t,x,v) + v \cdot \nabla_x f^{(1)}(t,x,v) + E^{(1)}(x) \cdot \nabla_v f^{(0)}(v) = 0.$$

Dynamics (e.g. for Landau damping $m = 1$)

$$f^{(0)}(v) + \sum_{k=1}^{m} \hat{f}_{k_i}^{(1)}(0,v) \mathrm{e}^{ik_i x} \xrightarrow{\text{time evolution}} f^{(0)}(v) + \sum_{k=1}^{m} \hat{f}_{k_i}^{(1)}(t,v) \mathrm{e}^{ik_i x}$$

which is at most rank $m+1$.

Reason 3: Certain limits have a low-rank structure

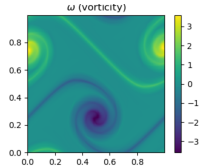
Collisional kinetic equations have a diffusive or fluid limit

$$\partial_t f(t,x,v) + v \cdot \nabla_x f(t,x,v) = \frac{1}{\epsilon}(f_{\text{eq}}(f) - f) \xrightarrow{\epsilon \to 0}$$

Euler equations

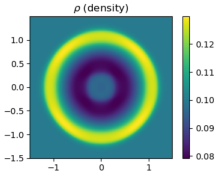
In the **incompressible case the limit is approximately low-rank**.

- Compressible case requires a different low-rank approximation.



Rigorous results available for linear Boltzmann equation.

- Based on a **Chapman–Enskog** expansion of the low-rank algorithm.



Z. Ding, L.E., Q. Li. SIAM J. Numer. Anal. 59(4) 2021.
L.E., J. Hu, L. Ying. SIAM J. Sci. Comput. 43(5) 2021.

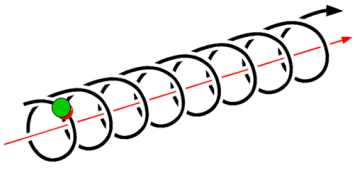
Why does low dynamical low-rank work?

In general, it is **not well understood under which conditions a kinetic problem admits a low-rank representation**.

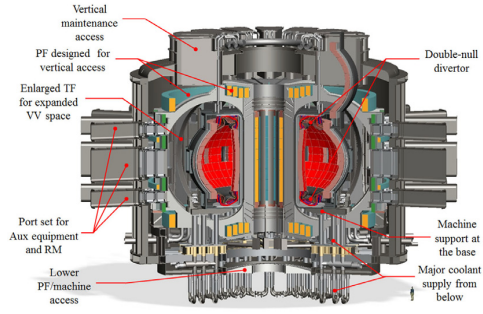
Alfvén waves

Strongly magnetized plasmas

In fusion applications we have **strongly magnetized** plasmas



$$F = q(E + \mathbf{v} \times \mathbf{B})$$



Gyrokinetics averages over the motion perpendicular to the magnetic fields.

- Reduces the problem to five dimensions (3 in space and 2 in velocity).
- Removes the extremely fast gyromotion from the model (order of ps for electrons).

Picture from doi:10.1088/0029-5515/55/5/053027 and Matthias Hirsch (CC).

A gyrokinetic model for shear Alfvén waves

Shear Alfvén waves are **electromagnetic** waves in a plasma that **propagate parallel to the magnetic field**.

- Little damping and thus important for stability.
- Complex interaction in tokamak due to toroidicity.

NASA MMS spacecraft study Alfvén waves in space.

Kinetic model with electric and magnetic fields for $f(t, x, y, z, v)$

$$\partial_t f + v\partial_z f + \frac{1}{M_e}\left(\partial_z \phi + \partial_t A\right)\partial_v f = 0, \qquad \Delta_\perp \phi = C_P \rho, \qquad \Delta_\perp A = C_A j.$$

with $\rho = 1 - \int f\, dv$ and $j = -\int vf\, dv$, $M_e = m_e/m_i$, $C_P = 1/(\rho_i/L)$ and $C_A = \beta C_P$.

Dynamical low-rank

Low-rank approximation based on **physical separation of motion along and perpendicular to magnetic** field

$$f(t,x,y,z,v) = \sum_{ij} X_i^f(t,x,y) S_{ij}^f(t) V_j^f(t,z,v).$$

but also for potentials ϕ and A

$$\phi(t,x,y,z) = \sum_{ij} X_i^\phi(t,x,y) S_{ij}^\phi(t) V_j^\phi(t,z),$$
$$A(t,x,y,z) = \sum_{ij} X_i^A(t,x,y) S_{ij}^A(t) V_j^A(t,z).$$

Advection treated exactly

$$\partial_t L_i^f + v\partial_z L_i^f + \frac{1}{M_e} \sum_{mk} (e_{ik}(z) + e_{ik}^A(z)) \partial_v L_k^f = 0, \qquad L_i = \sum_j S_{ij} V_j.$$

Reduced model

Reduced 1+1 dimensional model heavily employed by physicists assumes

$$f(t,x,y,z,v) = \hat{f}_{km}(t,z,v)\exp(ikx)\exp(imy).$$

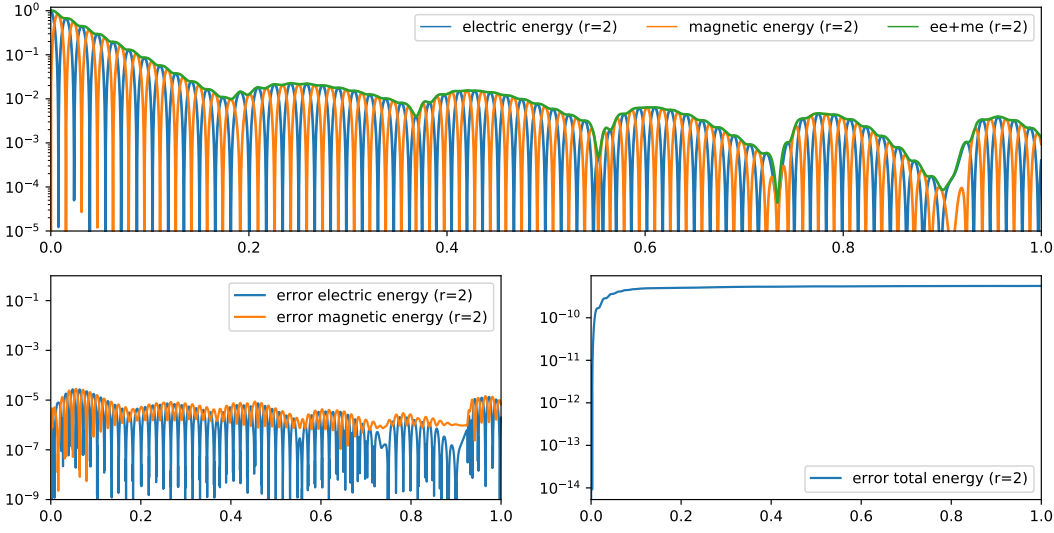
Within linear theory a dispersion relation can be derived

$$1 - \frac{2[1+\overline{\omega}Z(\overline{\omega})]}{(k_\perp \rho_i)^2}(\beta/M_e\overline{\omega}^2 - 1) = 0, \qquad \overline{\omega} = \omega/(v_{th,e}k_z).$$

The dynamical low-rank algorithm captures this solution exactly.

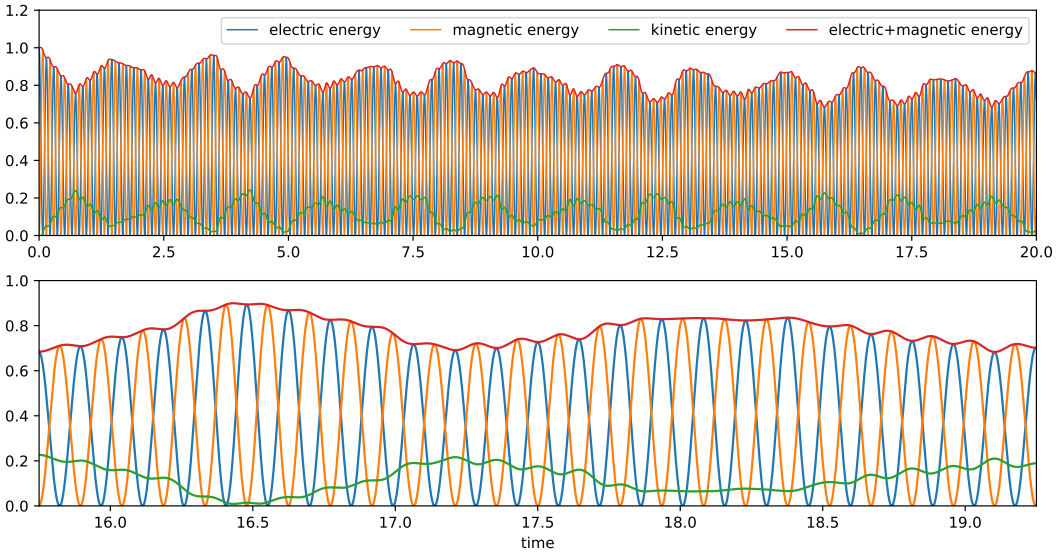
Verification

Error in the **Alfvén waves** simulation ($\alpha = 10^{-5}$, $k_\perp \rho_i = 0.2$, $\beta M_e = 2$).



Numerical simulation

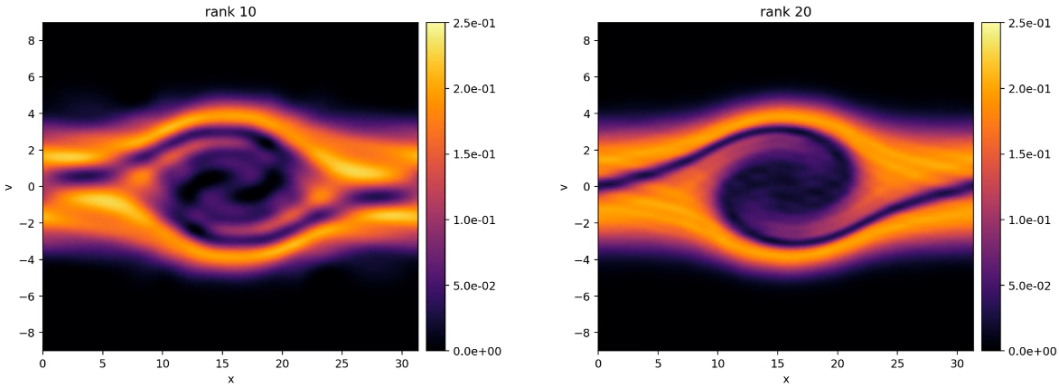
Alfvén wave simulation on a desktop computer, here with $r = 3$.



DLR is not just for the weakly nonlinear regime

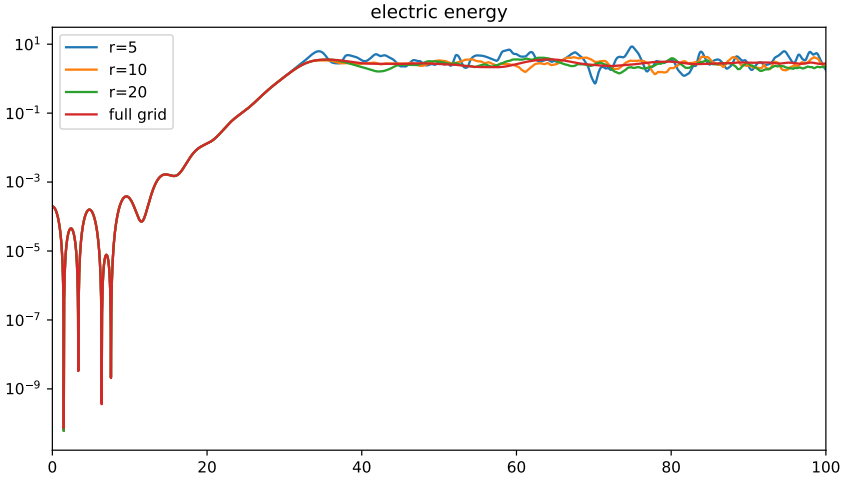
Two-stream instability

Low-rank approximation with 512 grid points per direction ($r = 10$ left, $r = 20$ right).



Two-stream instability

Time evolution of the electric energy.



Conservative dynamical low-rank approximation

Galerkin condition

Orthogonal projection

$$\text{Find} \quad \partial_t f = g \in T_f\mathcal{M} \quad \text{such that} \quad \|g - \text{RHS}\| \text{ is minimal.}$$

That is, $g = P(f)\text{RHS}$.

Galerkin condition

$$\text{Find} \quad \partial_t f \quad \text{such that} \quad \langle \nu, \partial_t f \rangle = \langle \nu, \text{RHS} \rangle \quad \forall \nu \in T_f\mathcal{M}.$$

For the Schrödinger equation implies symplecticity, energy, and norm conservation.

- But the situation here is very different.

L^2 conservation

Galerkin condition implies **L^2 norm conservation**

$$\partial_t \|f\|^2 = 2\langle f, \partial_t f\rangle_{xv} = 2\langle f, RHS\rangle_{xv} = 0$$

since $f \in T_f\mathcal{M}$.

But wait, why do we have $\langle f, RHS\rangle_{xv} = 0$?

This is how we (directly) prove L^2 conservation for the underlying model

$$\partial_t \|f\|^2 = 2\langle f, RHS\rangle_{xv} = \int -\nabla_x \cdot (vf^2) + \nabla_v \cdot (Ef^2)\,\mathrm{d}(x, v) = 0.$$

The analytic argument carries over. **This will be an important technique!**

Mass conservation

From

$$\partial_t f + \nabla_x \cdot (vf) - \nabla_v \cdot (Ef) = 0$$

we follow by integrating in v

$$\partial_t \int f \, \mathrm{d}x + \nabla_x \cdot \int vf \, \mathrm{d}v = 0,$$

which is more commonly written as

$$\partial_t \rho + \nabla \cdot j = 0, \qquad \rho = \int f \, \mathrm{d}v, \qquad j = \int vf \, \mathrm{d}v.$$

Integrating in x we get

$$M = \int f \mathrm{d}(x, v) = \text{const.}$$

That is, **conservation of mass**.

Momentum or charge conservation

From

$$\partial_t(vf) + \nabla_x \cdot ((v \otimes v)f) - v\nabla_v \cdot (Ef) = 0$$

we follow by integrating in v

$$\partial_t j + \nabla_x \cdot \sigma = -\int Ef \, \mathrm{d}v = E\rho, \qquad \sigma = \int (v \otimes v) f \, \mathrm{d}v.$$

Since

$$E(1-\rho) = \nabla \cdot (E \otimes E - \tfrac{1}{2}E^2)$$

and $\int E \, \mathrm{d}x = 0$ we obtain

$$P = \int vf \, \mathrm{d}(x, v) = \text{const},$$

That is, **conservation of momentum**.

Energy conservation

We already know that **energy (i.e. the Hamiltonian) is conserved**

$$H = \frac{1}{2}\int v^2 f \,\mathrm{d}(x,v) + \frac{1}{2}\int E^2 \,\mathrm{d}x.$$

Similar to mass and momentum there is also an associated **local conservation law**

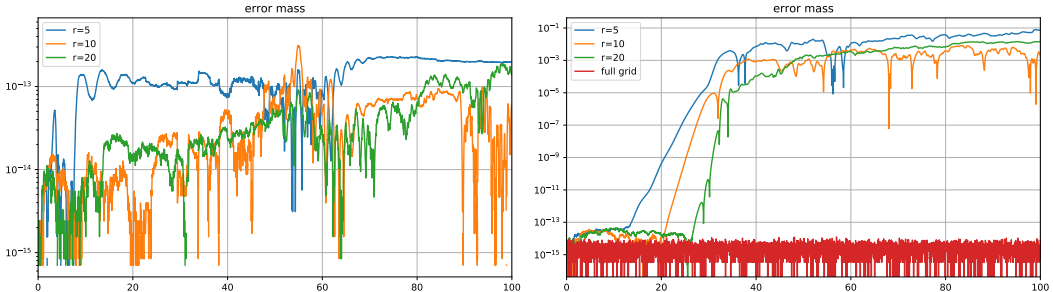
$$\partial_t e + \nabla_x \cdot Q = E \cdot (\partial_t E - j), \qquad e = \frac{1}{2}\int v^2 f \, dv + \frac{1}{2}E^2, \quad Q = \frac{1}{2}\int v v^2 f \, dv.$$

Dynamical low-rank approximation

The dynamical low-rank approximation finds the, in some sense, best L^2 approximation.

- No guarantee that mass, momentum, or energy is conserved.

Linear Landau damping (left) and two-stream instability (right).



This **failure is in stark contrast to Eulerian and particle methods**.

Literature

[Z. Peng, R. McClarren, M. Frank, J. Comput. Phys., 421 (2020)]

- Rescale solution to obtain mass conservation.
- Global mass conservation only.
- Not extensible to other invariants.

[Z. Peng, and R.G. McClarren. arXiv:2011.06072]

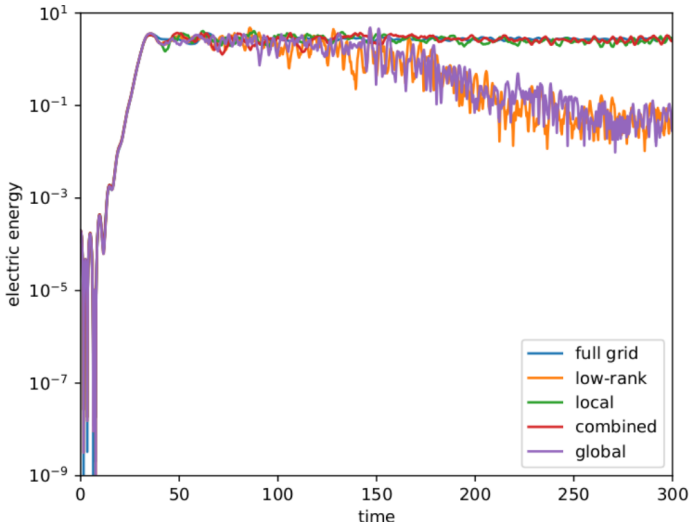
- Couple moments with low-rank approximation of g, where $f = M + g$.
- Needs to enforce $\int g \, \mathrm{d}(x, v) = 0$.
- Global invariants only.

[L. Einkemmer, C. Lubich. SIAM J. Sci. Comput., 40(5) (2018)]

- Add correction $\lambda_{ij} X_i V_j$ to enforce conservation (Lagrange multiplier).
- Conserves either global invariants or (a projected version of) conservation laws.
- Not able to simultaneously conserve both.

Global vs local conservation

We should not forget the **local conservation law!**



Conservative dynamical low-rank approximation

Conservative dynamical low-rank approximation

Fundamental observation: If $v \mapsto 1/v \mapsto v/v \mapsto v^2$ is part of the approximation space $\overline{V} = \text{span}\{V_1, \dots, V_r\}$ then we obtain the conservation laws also in the DLRA.

For $K_j = \sum_i X_i S_{ij}$ we have

$$f = \sum_j K_j V_j, \qquad \text{and thus} \qquad \rho = \int f \, \mathrm{d}v = \sum_j K_j \langle 1, V_j \rangle_v.$$

Now we assume that $V_1 \propto 1$. Then

$$\rho = \frac{1}{V_1} K_1$$

and thus

$$\partial_t \rho = \frac{1}{V_1} \langle V_1, \text{RHS} \rangle_v = \int_{\Omega_v} \text{RHS} \, dv = -\nabla \cdot j.$$

Argument from the continuous system carries over.

Problems

Problem 1: These functions do not lie in $L^2(\mathbb{R}^3)$.

- We use an $\mathbf{L}^2$ **space weighted by** f_{0v}.
- For kinetic equations $f_{0v}(v) = exp(-v^2/2)$ is usually a reasonable choice.

Low-rank approximation

$$f = f_{0v} \sum_{ij} X_i S_{ij} V_j$$

with $X_i \in L^2(\Omega_x)$ and

$$V_j \in L^2(\Omega_v, f_{0v}) = \left\{ g \colon \int f_{0v} g^2 \, \mathrm{d}x < \infty \right\}.$$

Problem 2: The basis functions are chosen by the algorithm to satisfy a Galerkin condition.

- Basis functions **change as time evolves in order to adapt to the problem**.

Conservative dynamical low-rank algorithm

Some of the v dependent basis functions V_j are held fixed

$$U_a(v) = V_a(t, v), \quad 1 \le a \le m \qquad \text{and} \qquad W_p(t, v) = V_p(t, v), \quad m < p \le r.$$

But orthogonality between U_a and W_p still **needs to be enforced.**

Petrov–Galerkin condition

$$\left(\frac{\nu}{f_{0v}}, \partial_t f - RHS \right)_{xv} = 0 \qquad \forall \nu \in T_f \mathcal{M}$$

with $(f, g) = \int_{\Omega_v} fg \, \mathrm{d}v$.

Equations of motion for $\mathbf{S}_{ij}$: We test with $\nu_{kl} = f_{0v} X_k V_l$

Equations of motion for X

We test with $\nu_k = f_{0v}\chi V_k$, χ is arbitrary.

- Since $\nu_k = f_{0v}\sum_{ij}\dot{X}_i S_{ij} V_j$ with $\dot{X}_i = \chi(x) S_{ki}^{-1}$ it holds that $\nu_k \in T_f\mathcal{M}$.

The Petrov–Galerkin condition becomes

$$\left(V_k\chi, f_{0v}\sum_{ij}\left(\dot{X}_i S_{ij} V_j + X_i \dot{S}_{ij} V_j\right) + f_{0v}\sum_{ip} X_i S_{ip}\dot{W}_p\right)_{xv} = (V_k\chi, \mathrm{RHS})_{xv}\,.$$

which we can rewrite as

$$\left\langle V_k\chi, \sum_{ij}\left(\dot{X}_i S_{ij} V_j + X_i \dot{S}_{ij} V_j\right) + \sum_{ip} X_i S_{ip}\dot{W}_p\right\rangle_{xv} = (V_k\chi, \mathrm{RHS})_{xv}$$

Using orthogonality/gauge cond. and χ arbitrary, we obtain the **equations of motion**

$$\sum_i \dot{X}_i S_{ik} = (V_k, \mathrm{RHS})_v - \sum_i X_i \dot{S}_{ik}.$$

Equations of motion for W

We test with $\nu_q = f_{0v}\zeta \sum_i X_i S_{iq}$, ζ is arbitrary.

- Since $\nu_q = f_{0v} \sum_{ip} X_i S_{ip} \dot{W}_p$ with $\dot{W}_p = \delta_{pq}\zeta(v)$ it holds that $\nu_q \in T_f\mathcal{M}$.

The Petrov–Galerkin condition becomes

$$\sum_i \left(\zeta X_i S_{iq}, f_{0v} \sum_{kl} \left(\dot{X}_k S_{kl} V_l + X_k \dot{S}_{kl} V_l \right) + f_{0v} \sum_{kp} X_k S_{kp} \dot{W}_p \right)_{xv} = \sum_i (\zeta X_i S_{iq}, \mathrm{RHS})_{xv}$$

Using orthogonality/gauge cond. and ζ arbitrary, we obtain the **equations of motion**

$$\sum_{ip} S_{iq} S_{ip} (\partial_t W_p) + \sum_{il} S_{iq} (\partial_t S_{il}) V_l = \frac{1}{f_{0v}} \sum_i S_{iq} (X_i, \mathrm{RHS})_x.$$

Coefficients

The **coefficients are slightly different** due to the weighted approximation space.

For example

$$\begin{aligned}
\frac{1}{f_{0v}}(X_i, \mathrm{RHS})_x &= \frac{1}{f_{0v}}\left(X_i, -v\cdot\nabla_x f + E\cdot\nabla_v f\right)_x \\
&= -\frac{1}{f_{0v}}\sum_{kl} f_{0v}\langle X_i, \nabla_x X_k\rangle_x \cdot v S_{kl} V_l + \frac{1}{f_{0v}}\sum_{kl} S_{kl}\nabla_v(f_{0v}V_l)\cdot\langle X_i, EX_k\rangle_x \\
&= -\sum_{kl}(v\cdot d^2_{ik})S_{kl}V_l + \frac{1}{f_{0v}}\sum_{kl} d^1_{ik}[E]\cdot\nabla_v(f_{0v}S_{kl}V_l) \\
&= -\sum_{kl}(v\cdot d^2_{ik})S_{kl}V_l + \sum_{kl} d^1_{ik}[E]\cdot\left[\nabla_v(S_{kl}V_l) + \nabla_v(\log f_{0v})S_{kl}V_l\right],
\end{aligned}$$

where

$$d^1_{ik}[E] = \langle X_i E X_k\rangle_x, \qquad d^2_{ik} = \langle X_i, \nabla_x X_k\rangle_x.$$

Conservative dynamical low-rank algorithm

We have

- $U_1 \propto 1$, $U_2 \propto v$, $U_3 \propto v^2$ lie in the approximation space span$\{V_1, \dots, V_r\}$ by construction;
- The dynamics determined W_p are orthogonal to the U_a as

$$\begin{aligned}\partial_t \langle U_a, W_p \rangle_v &= \sum_{iq} T^{-1}_{pq} S_{iq} \langle \tfrac{1}{f_{0v}} U_a X_i, \mathrm{RHS} \rangle_{xv} - \sum_{il} T^{-1}_{pq} S_{iq} \partial_t S_{il} \langle U_a, V_l \rangle_v \\ &= \sum_{iq} T^{-1}_{pq} S_{iq} \langle \tfrac{1}{f_{0v}} U_a X_i, \mathrm{RHS} \rangle_{xv} - \sum_{il} T^{-1}_{pq} S_{iq} (X_i U_a, \mathrm{RHS})_{xv} = 0.\end{aligned}$$

Results in a **mass, momentum, and energy conservative** DLR approximation.

Momentum conservation

We choose U_2 such that $v = \|v\| U_2$, i.e. $U_2 \propto v$.

Our **dynamical low-rank approximation is conservative** because we can use the **argument for the original problem**.

For example, for the **momentum density** we have

$$j = \int vf \,\mathrm{d}v = \sum_j K_j \int f_{0v} v V_j \,\mathrm{d}v = \|v\| \sum_j K_j \langle U_2, V_j \rangle = \|v\| K_2.$$

Conservation of momentum

$$\partial_t j = \|v\| \partial_t K_2 = \|v\| (U_2, \mathrm{RHS})_v = \int v \mathrm{RHS}\, dv = -\nabla_x \cdot \sigma - E\rho.$$

Integration in x then gives the global invariant.

Conservative time and space discretization

Time discretization

Explicit Euler scheme applied to the equations of motion

$$S_{kl}^{n+1} = S_{kl}^{n} + \Delta t \, (X_k^n V_l^n, \mathrm{RHS}^n)_{xv} \, ,$$

$$X_i^{n+1} = X_i^n + \Delta t \sum_k (S^n)_{ik}^{-1} \left[(V_k^n, \mathrm{RHS}^n)_v - \sum_l X_l^n \, (X_l^n V_k^n, \mathrm{RHS}^n)_{xv} \right],$$

$$W_p^{n+1} = W_p^n + \Delta t \sum_q ((S^n)^T S^n)_{pq}^{-1} \left[\frac{1}{f_{0v}} \sum_i S_{iq}^n (X_i^n, \mathrm{RHS}^n)_x - \sum_{il} S_{iq}^n \, (X_i^n V_l^n, \mathrm{RHS}^n)_{xv} \, V_l^n \right],$$

is **not** conservative.

Uses S^n, i.e. S at time t^n, to compute X^{n+1}.

- There is no well defined K^n and K^{n+1} and thus the argument applied before does not carry over.

Conservative time discretization

We can rewrite the equation for K in conservative form

$$\partial_t \left(\sum_i X_i S_{ik} \right) = \left(V_k, \mathrm{RHS} \right)_v .$$

Discretization yields the **conservative Euler** scheme

$$\begin{aligned}
S_{kl}^{n+1} &= S_{kl}^n + \Delta t \left(X_k^n V_l^n, \mathrm{RHS}^n \right)_{xv} , \\
X_i^{n+1} &= \sum_k (S^{n+1})_{ik}^{-1} \left[\sum_j X_j^n S_{jk}^n + \Delta t \left(V_k^n, \mathrm{RHS}^n \right)_v \right], \\
W_p^{n+1} &= W_p^n + \Delta t \sum_{qi} ((S^n)^T S^n)_{pq}^{-1} S_{iq}^n \left[\frac{1}{f_{0v}} (X_i^n, \mathrm{RHS}^n)_x - \sum_l \left(X_i^n V_l^n, \mathrm{RHS}^n \right)_{xv} V_l^n \right].
\end{aligned}$$

Method is **fully explicit** and **mass and momentum conservative up to machine precision**.

Conservation laws

The **conservative Euler scheme** also satisfies the **discretized versions of the local conservation laws**.

Mass:

$$\frac{\rho^{n+1}-\rho^n}{\Delta t} = \frac{1}{U_1}\frac{K_1^{n+1}-K_1^n}{\Delta t} = \int_{\Omega_v} \mathrm{RHS}^n \, dv = -\nabla_x \cdot j^n.$$

Momentum:

$$\frac{j^{n+1}-j^n}{\Delta t} = \|v\|\frac{K_2^{n+1}-K_2^n}{\Delta t} = \int_{\Omega_v} v\mathrm{RHS}^n \, dv = -\nabla_x \cdot \sigma^n - E^n\rho^n.$$

Failure of energy conservation

We choose U_3 such that $v^2 - 1 = \|v^2 - 1\| U_3$, i.e. $U_3 \propto v^2 - 1$.

We have

$$\begin{aligned}\frac{e^{n+1} - e^n}{\Delta t} &= \|v^2 - 1\| \frac{K_3^{n+1} - K_3^n}{2\Delta t} + \|1\| \frac{K_1^{n+1} - K_1^n}{2\Delta t} + \frac{(E^{n+1})^2 - (E^n)^2}{2\Delta t} \\ &= \frac{1}{2} \int_{\Omega_v} v^2 \mathrm{RHS}^n \, dv + E^n \frac{E^{n+1} - E^n}{\Delta t} + \frac{(E^{n+1} - E^n)^2}{2\Delta t} \\ &= \nabla_x \cdot Q^n + E^n \cdot \left(\frac{E^{n+1} - E^n}{\Delta t} - j^n \right) + \frac{(E^{n+1} - E^n)^2}{2\Delta t}.\end{aligned}$$

Integrating in x yields

$$H^{n+1} - H^n = \Delta t \int_{\Omega_x} E^n \cdot \left(\frac{E^{n+1} - E^n}{\Delta t} - j^n \right) dx + \frac{1}{2} \int_{\Omega_x} (E^{n+1} - E^n)^2 \, \mathrm{d}x = \mathcal{O}(\Delta t^2).$$

Energy conservation

Can be remedied by solving

$$\partial_t f + v\nabla_x f - E^{n+1/2}\nabla_v f = 0, \qquad E^{n+1/2} = (E^{n+1} + E^n)/2.$$

Resulting scheme is energy conservative

$$\begin{aligned}\frac{e^{n+1} - e^n}{\Delta t} &= \nabla_x \cdot Q^n - E^{n+1/2} \cdot j^n + \frac{(E^{n+1} - E^n)(E^{n+1} + E^n)}{2\Delta t} \\ &= \nabla_x \cdot Q^n + E^{n+1/2} \cdot \left(\frac{E^{n+1} - E^n}{\tau} - j^n \right)\end{aligned}$$

but **implicit**.

Conservative space discretization

Obtaining a **conservative space discretization is straightforward**.

- Assumption on the method is that discrete integration by parts is exact.

Examples

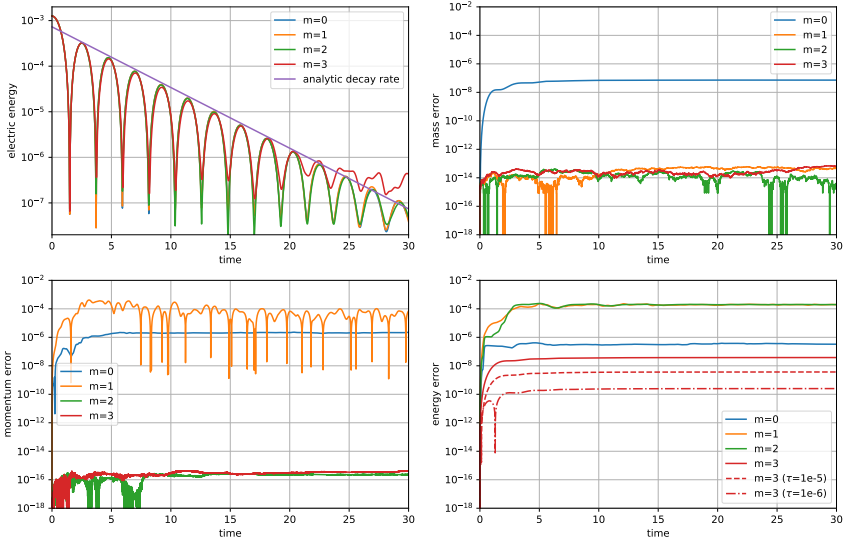
- FFT based methods
- Standard second-order centered finite differences
- discontinuous Galerkin schemes with centered flux

Discrete integration by parts for centered differences and periodic boundary conditions

$$\sum_{i=0}^{n-1}(g_{i+1} - g_{i-1}) = \sum_{i=1}^{n} g_i - \sum_{i=-1}^{n-2} g_i = g_n - g_0 + g_{n-1} - g_{-1} = 0.$$

Numerical results

Conservative DLR and (fully explicit) conservative Euler scheme.



Robustness

Robustness to overapproximation is desirable (especially if rank adaptivity is used).

- Avoid inverting S.

Conservative DLR algorithm can not be combined with projector splitting.

Unconventional integrator

Step 1: Compute K_k^{n+1} and $L_q^{n+1,n}$ with $K_k^{n+1} = \sum_i X_i^{n+1} S_{ik}^{n+1}$ and $L_q^{n+1,n} = \sum_{ip} S_{iq}^n S_{ip}^n W_p^{n+1}$.

Step 2: Perform a QR decomposition

$$K_k^{n+1} = \sum_i X_i^{n+1} R_{ik}^1, \qquad L_q^{n+1,n} = \sum_p W_p^{n+1} R_{pq}^2.$$

and throw away R_{ik}^1 and $L_q^{n+1,n}$.

Step 3: Find the best approximation in $\overline{X} = \text{span}\{X_i\}$ and $\overline{V} = \text{span}\{V_j\}$

$$S_{kl}^{n+1} = \sum_{ij} M_{ki} S_{ij}^n N_{jl}^T + \Delta t \left(X_k^{n+1} V_l^{n+1}, \text{RHS}[f(X^{n+1}, MS^n N^T, V^{n+1})] \right)_{xv},$$

$$M_{ki} = \langle X_k^{n+1} X_i^n \rangle_x, \qquad N_{jl}^T = \langle V_j^n V_l^{n+1} \rangle_v.$$

G. Ceruti, C. Lubich. BIT Numer. Math. 62 (2022).

Unconventional integrator

The unconventional integrator does destroy conservation.

The projection given by

$$\sum_k X_k^{n+1} M_{ki} = \sum_k \langle X_i^n, X_k^{n+1} \rangle_x X_k^{n+1} = P_{\overline{X^{n+1}}} X_i^n.$$

is not exact.

In order to **preserve the local conservation laws** we need to satisfy, e.g.,

$$\partial_t \rho + \nabla_x \cdot j = 0.$$

However, $\nabla_x \cdot j$ does not necessarily lie in $\overline{X^n}$.

Conservative unconventional integrator

In order to make the unconventional integrator conservative we make the following two modifications.

Modification 1: Augment the basis to

$$\begin{bmatrix} K^{n+1} & X^n & \nabla X^n \end{bmatrix} \qquad \text{and} \qquad \begin{bmatrix} U & L^{n+1} & W^n \end{bmatrix}.$$

This would increase the rank in each time step.

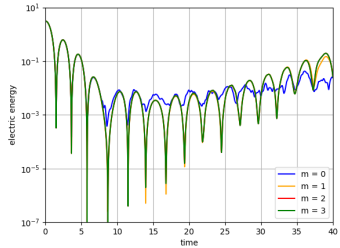
Modification 2: Perform a truncation (projection) to rank r.

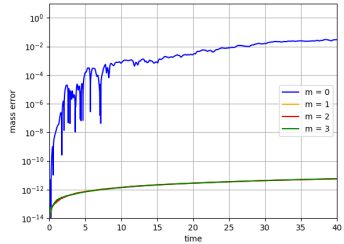
- This has to be done such that the projection on U is exact.

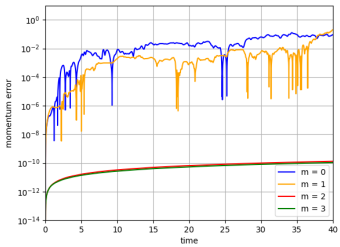
We first project onto U (without error) and then truncate the remainder using a SVD.

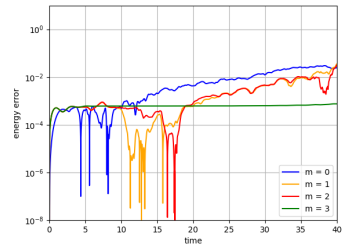
Numerical results

Nonlinear Landau damping with $r = 25$.









Literature

Literature

[O. Koch, C. Lubich. SIAM J. Matrix Anal. Appl., 29(2), 2007]

- Dynamical low-rank algorithm for ODEs in the matrix case.

[C. Lubich, I.V. Oseledets. BIT Numer. Math. 54(1) 2014]

- Projector splitting to obtain a robust integrators (ODE case).

[L.E., C. Lubich, SIAM J. Sci. Comput. 40(5), 2018]

- Projector splitting based dynamical low-rank algorithm for Vlasov–Poisson.
- Probably the best starting point to get more details.

[L.E., A. Ostermann, C. Piazzola, J. Comput. Phys. 403, 2020]

- Extension to Vlasov–Maxwell which respects the divergence constraint.
- Low-rank structure for the linear Vlasov–Maxwell equations.

Literature

[L.E., I. Joseph. J. Comput. Phys. 443, 2021]

- The conservative dynamical low-rank method.

[J. Kusch, G. Ceruti, L.E., M. Frank, arXiv:2105.04358]

- Related ideas for enforcing boundary conditions in uncertainty quantification.

[Z. Ding, L.E., Q. Li. SIAM J. Numer. Anal, 59(4), 2021]

- Analysis of dynamical low-rank in the diffusive limit.
- Fully implicit scheme.

[L.E., J. Hu, Y. Wang. J. Comput. Phys. 439, 2021]

- IMEX based efficient numerical methods for the diffusive limit.
- Preserves the AP property of the fully implicit scheme.

[L.E., J. Hu, L. Ying. arXiv:2101.07104]

- AP dynamical low-rank scheme for the compressible Navier–Stokes limit.